UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/693,921	10/28/2003	Kazuhiro Hosotani	117630	3993

25944 7590 05/09/2008

OLIFF & BERRIDGE, PLC
P.O. BOX 320850
ALEXANDRIA, VA 22320-4850

EXAMINER	
CHENG, PETER L	

ART UNIT	PAPER NUMBER
2625	

MAIL DATE	DELIVERY MODE
05/09/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No. 10/693,921	Applicant(s) HOSOTANI ET AL.	
	Examiner PETER L. CHENG	Art Unit 2625	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE _3_ MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on _20 December 2007_.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) _1-8_ is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) _1-8_ is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on _01 March 2004_ is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Response to Arguments

1. Applicant's arguments, see **page 10, lines 15 - 17**, filed 12/20/2007, with respect to the rejections of claims 1, 2 and 6 – 8 under 35 U.S.C. 102(b) have been fully considered and are persuasive. Therefore, the rejection has been withdrawn. However, upon further consideration, a new ground of rejection is made in view of **HANSEN [US Patent 7,185,014 B1]**. As noted in the rejection below, HANSEN teaches a method for determining that the model numbers are the same.

Drawings

2. The drawings are objected to because:

- **Fig. 6:** regarding step **S67**, it is assumed that applicant intended to cite

  MODEL INFORMATION (APPLIANCE NAME AND VERSION) IS <u>MODEL</u> COINCIDENT WITH STORED MODEL INFORMATION?

  instead of

  MODEL INFORMATION (APPLIANCE NAME AND VERSION) IS <u>MODE</u> COINCIDENT WITH STORED MODEL INFORMATION?;

Corrected drawing sheets in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. The figure or figure number of an amended drawing should not be labeled as “amended.” If a drawing figure is to be canceled, the appropriate figure must be removed from the replacement sheet, and where necessary, the remaining figures must be renumbered and appropriate changes made to the brief description of the several views of the drawings for consistency. Additional replacement sheets may be necessary to show the renumbering of the remaining figures. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either “Replacement Sheet” or “New Sheet” pursuant to 37 CFR 1.121(d). If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Specification

3. The disclosure is objected to because of the following informalities; please note that page and line numbers refer to the originally filed specification:

- There are some typographical and grammatical errors in the disclosure; if claims were to be subsequently allowed, the following concerns should be

addressed; for example, **page 2, line 5 (**replace "is to be needed" with "is needed"**); page 2, line 14 (**insert "an" between "that" and "erroneous"**); page 14, line 18 (**insert "a" between "with" and "configuration"**); page 16, line 14 (**replace "an model name" with "a model name"**); page 18, line 6 (**replace "a cold reserving storage" with the more common term "refrigerator", as noted on page 11, line 8**); page 19, line 8 (**replace "an model name" with "a model name"**); page 26, line 13 (**insert "an" before the first "editing"**); page 26, line 22 (**replace "an model name" with "a model name"**); page 27, lines 22 – 23 (**suggest replacing "set to the first printer 2" with "set to a first printer 2"**); page 27, lines 23 – 24 (**suggest replacing "set to the next printer 2" with "set to a second printer 2"**); page 27, lines 24 – 25 (**suggest replacing "set to the *further next* printer 2" with "set to a third printer 2"**); page 28, line 7 (**replace "an model name" with "a model name"**); page 30, line 7 (**replace "a erroneous" with "an erroneous"**); page 30, line 25 (**replace "may case" with "may cause"**); page 31, line 8 (**replace "fourthe" with "fourth"**); page 32, line 23** and **page 35, line 19 (**suggest changing "the erroneous operation" to "an erroneous operation"**); page 33, line 13** and **page 36, line 8 (**suggest "if a configuring error occurs" instead of "when the configuring error happens to occur"**); page 33, line 14** and **page 36, line 10 (**suggest "determine" or similar word instead of "find out"**);**

- **Page 14, line 17:** reference number **15** is associated with a "ROM" *(read-only memory)*; however, on **page 14, line 6,** reference number **15** is associated with a "storage unit" *such as a hard disk drive*; since these are different devices, for clarity, suggest removing reference number **15** after the word "ROM";

- **Page 14, line 19:** for clarity, suggest adding reference number "151" after "file"; that is, "information storage file **151** which stores …";

- **Page 18, line 18:** for clarity, suggest replacing **machine sort file 252** with ***model information* file 252**;

- **Page 26, line 17:** for clarity, suggest replacing **a column for entering model information** with **an area for entering model information**;

- **Page 26, line 18:** for clarity, suggest replacing **another column for entering a configuring item** with **another area for entering a configuring item**;

Appropriate correction is required.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. The factual inquiries set forth in *Graham* **v.** *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

6. Claims 1, 2, 3, 4, 6, 7 and 8 are rejected under 35 U.S.C. 103(a) as being unpatentable over **GOFFINET [US Patent 5,905,906]** in view of **HANSEN [US Patent 7,185,014 B1]**.

As for claim 1, GOFFINET teaches a device configuring method for configuring a plurality of devices of various kinds by an information processing apparatus [**Fig. 1,** host computer **12**] which is connected to the devices via a communication network [**Fig. 1** LAN **15**], the method comprising:

acquiring from a first device [Fig. 1 printer **13] both model information of the first device and identification information specific to the first device by the information processing apparatus**

[GOFFINET teaches a method by which a host computer (**Fig. 1** host computer **12**) may "save the configuration information of a particular printer (e.g., printer 13)"; **col. 6, lines 47 - 48.** This is illustrated in **Fig. 4** as the "Quick Setup Save" procedure.

Within the printer controller (**Fig. 3**), the "Options Manager 37 is designed to be able to easily retrieve and store all such configuration variables for its particular model laser printer"; **col. 7, lines 5 – 7.**

Shown in **Fig. 4,** steps **102 (**Determine I.D. of Next OM Variable to be Read**)** through **110 (**At End of OM Table?**),** the Options Manager reads each OM variable shown in **Table #1** and transmits the corresponding value to the host computer where it is stored in a file.

From **Table #1,** the host computer acquires "model information" of a first device (e.g., OMMODELNAME, the model name) and "identification information" (e.g., OMSERIALNUM, the serial number);

other related "model information" could be obtained from Table #1 as well; for example, some "higher-end" models may have 3 paper input trays whereas, some "lower-end" models may just have one**];**

acquiring from the first device configuration information of the first device by the information processing apparatus

[Contained within **Table #1** are various configuration variables (i.e., "OM variables"); among these are, for example, **OMEMULATION** (default emulation), **OMPAPERSRC** (default paper source), **OMOUTPUTCAP** (output drawer capacity), and **OMINPUTCAP** (input tray1 capacity)**];**

storing the acquired configuration information in a status correlated with both the model information and the identification information of the first device

[Fig. 4 "depicts a flow chart of the steps that the host computer (e.g., a host 12) must undergo to create a file at its own storage media (e.g., upon its own hard disk drive) so as to save the configuration information of a particular printer"; **col. 6, lines 44 – 47;** the storing of the acquired configuration information occurs in **Fig. 4** step **112** (close file in which printer settings are stored)**];**

acquiring from a second device [e.g., printer **16a** shown in **Fig. 1] both model information of the second device and identification information specific to the second device by the information processing apparatus**

[As with the first device (i.e., printer **13**), the host computer could obtain both model and identification information specific to a second device, and display model information as shown in **Fig. 8;** "those printers having bi-directional communications capabilities are noted with an asterisk, such as that indicated by index numeral 212"; **col. 22, lines 12 - 14];**

~~determining~~ ***<u>causing the information processing apparatus or the second device to determine</u>*** **whether or not the model information of the first device and the model information of the second device coincide with each other**

[GOFFINET teaches that once the model, identification and configuration information are acquired from the first device and are stored in a file (**Fig. 4** step **112**), "the file on the hard drive can be accessed and its contents sent to other printers on the LAN 15, thereby configuring such other printers very quickly and easily"; **col. 15, lines 3 – 6.**

GOFFINET further teaches that "under normal circumstances, it is preferred that such setup or configuration information for a particular printer be utilized on other printers having the identical model number"; **col. 15, lines 6 - 9];**

transmitting, when determined that the model information of the first device and the model information of the second device coincide with each other, the stored configuration information of the first device from the information processing apparatus to the second device

[As noted above, GOFFINET teaches that it is preferable that the first and second devices have the "identical model number".

Fig. 6 illustrates the "Quick Setup Send" host computer procedure. Once a configuration setup file has been selected (**step 140**), a selection is made as to which printers will be configured (**step 142**). **Steps 144** through **152** retrieve configuration values stored in the saved setup file and transmit each value (along with its corresponding variable identification) to a second device (i.e., a selected printer); specifically, the data packet for a "Set OM Variable" command has a format shown in **col. 15, line 29];**

and configuring the second device in accordance with the transmitted configuration information

[Fig. 7 illustrates the "Set OM Variable" printer procedure. After the second device (i.e., a selected printer) receives the packet sent from the host computer, the printer controller's Options Manager reads the value of the OM variable (from the data packet) and stores it into memory; **col. 16, lines 32 – 35.**

"Configuring the second device" is achieved by storing the new OM variable values into memory**].**

However, as noted by applicant's remarks filed on 12/20/2007 on **page 11, lines 2 – 4**, GOFFINET *does not disclose or suggest any method or mechanism for determining that the model numbers are the same*.

HANSEN teaches a system and method of retrieving data from a server and cites, "A system includes a server and a controller embedded in a device. Both the server and the embedded controller are capable of communicating over a computer network. The embedded controller sends a command to the server over the computer network that identifies *an instance of the device*. In response, the server identifies the *instance of the device* based on the command, retrieves data that is specific to *the instance of the device*, and sends the data to the embedded controller over the computer network"; **abstract.**

With reference to **Fig. 2**, HANSEN defines "*instance*" of a device as "the specific identity of device 11 as distinguished from other identical devices. The identification information stored in database 24 identifies the instance of device 11. This identification information includes, but is not limited to, data identifying the type of the device, a common (or "friendly") name for the device, the manufacturer of the device, *the model name of the*

device, the model number of the device, the serial number of the device, and a universal unique identifier (UUID) for the device"; **col. 3, lines 33 – 42**.

In addition to storing "identification information", the embedded controller's database **24** stores "operational parameters" and "configuration files"; **col. 3, lines 19 – 22.** "Operational parameters constitute settings and /or control instructions for the device 11, which are implemented by embedded controller 17"; **col. 3, lines 23 – 24.** "A configuration file is a file that contains a set of one or more operational parameters for an instance of device 11"; **col. 3, lines 29 – 31.**

With reference to **Fig. 1**, the server's database **30** stores *operational parameters* "individually or as part of a configuration file for an instance of device 11"; **col. 4, lines 17 – 19.**

With reference to **Fig. 2,** the "embedded controller 17 executes software 20 to retrieve data, such as operational parameters, from remote server 19"; **col. 4, lines 27 – 28.** In step **201**, the embedded controller [i.e., a "second device"] sends a command to the server [i.e., an "information processing apparatus"] which "includes data for identifying device 11. The data identifies the specific instance of device 11 and includes a device type field and one or both of a device serial number field and a device UUID. The command may also include the common name field, the manufacturer name field, *the model name field*, and the model number field"; **col. 4, lines 42 – 46.**

In step **202**, the server “receives the command from embedded controller 17” [**col. 5, lines 39 - 40**] and in step **203**, the server parses “the various identifying fields”; **col. 5, lines 45 – 47.** In step **204**, the server “identifies the instance of device 11 based on the information parsed from the command”; **col. 5, lines 48 – 49.** Referring to **Fig. 5**, in step **205**, data which is specific to the instance of the device is retrieved, and in step **206**, the retrieved data is sent to the embedded controller.

By sending identification information from the embedded controller (i.e., a “second device”) to a server (i.e., an “information processing apparatus”), and causing the server to identify the instance of the device (containing the embedded controller) based on the identification information, HANSEN teaches a method and mechanism for determining that the model numbers are the same.

It would have been obvious to one of ordinary skill in the art at the time the invention was made combine the teachings of HANSEN with those of GOFFINET so that the information processing apparatus could *determine whether or not the model information of the first device and the model information of the second device coincide with each other.*

Regarding claim 2, GOFFINET teaches a device configuring system comprising:

a plurality of devices of various kinds

[GOFFINET teaches “each of the printers 13, 16a, 16b, 16c, and 16d may be of the same type or of different models”; **col. 3, lines 57 - 59];**

and an information processing apparatus [**Fig. 1,** host computer **12**] **which is connected to the devices via a communication network** [**Fig. 1** LAN **15**], **wherein the information processing apparatus comprises:**

a first acquiring unit configured to acquire from a first device [Fig. 1 printer **13] both model information of the first device and identification information specific to the first device**

[GOFFINET teaches a method by which a host computer (**Fig. 1** host computer **12**) may “save the configuration information of a particular printer (e.g., printer 13)”; **col. 6, lines 47 - 48.** This is illustrated in **Fig. 4** as the “Quick Setup Save” procedure.

Within the printer controller (**Fig. 3**), the “Options Manager 37 is designed to be able to easily retrieve and store all such configuration variables for its particular model laser printer”; **col. 7, lines 5 – 7.**

Shown in **Fig. 4,** steps **102 (**Determine I.D. of Next OM Variable to be Read**)** through **110 (**At End of OM Table?**),** the Options Manager reads each OM

variable shown in **Table #1** and transmits the corresponding value to the host computer where it is stored in a file.

From **Table #1,** the host computer acquires “model information” of a first device (e.g., OMMODELNAME, the model name) and “identification information” (e.g., OMSERIALNUM, the serial number);

other related “model information” could be obtained from Table #1 as well; for example, some “higher-end” models may have 3 paper input trays whereas, some “lower-end” models may just have one**];**

a configuration information acquiring unit configured to acquire from the first device configuration information of the first device

[Contained within **Table #1** are various configuration variables (i.e., “OM variables”); among these are, for example, **OMEMULATION** (default emulation), **OMPAPERSRC** (default paper source), **OMOUTPUTCAP** (output drawer capacity), and **OMINPUTCAP** (input tray1 capacity)**];**

a storing unit configured to store the acquired configuration information in a status correlated with both the model information and the identification information of the first device

[Fig. 4 "depicts a flow chart of the steps that the host computer (e.g., a host 12) must undergo to create a file at its own storage media (e.g., upon its own hard disk drive) so as to save the configuration information of a particular printer"; **col. 6, lines 44 – 47;** the storing of the acquired configuration information occurs in **Fig. 4** step **112** (close file in which printer settings are stored)**];**

a second acquiring unit configured to acquire from a second device [e.g., printer **16a** shown in **Fig. 1] both model information of the second device and identification information specific to the second device**
[As with the first device (i.e., printer **13**), the host computer could obtain both model and identification information specific to a second device, and display model information as shown in **Fig. 8;** "those printers having bi-directional communications capabilities are noted with an asterisk, such as that indicated by index numeral 212"; **col. 22, lines 12 - 14];**

a determining unit configured to determine whether or not the model information of the first device and the model information of the second device coincide with each other
[GOFFINET teaches that once the model, identification and configuration information are acquired from the first device and are stored in a file (**Fig. 4** step **112**), "the file on the hard drive can be accessed and its contents sent to other

printers on the LAN 15, thereby configuring such other printers very quickly and easily"; **col. 15, lines 3 – 6.**

GOFFINET further teaches that "under normal circumstances, it is preferred that such setup or configuration information for a particular printer be utilized on other printers having the identical model number"; **col. 15, lines 6 - 9];**

and a transmitting unit configured to transmit, when determined that the model information of the first device and the model information of the second device coincide with each other, the stored configuration information of the first device to the second device
[As noted above, GOFFINET teaches that it is preferable that the first and second devices have the "identical model number".

Fig. 6 illustrates the "Quick Setup Send" host computer procedure. Once a configuration setup file has been selected (**step 140**), a selection is made as to which printers will be configured (**step 142**). **Steps 144** through **152** retrieve configuration values stored in the saved setup file and transmit each value (along with its corresponding variable identification) to a second device (i.e., a selected printer); specifically, the data packet for a "Set OM Variable" command has a format shown in **col. 15, line 29],**

wherein the second device comprises a configuring unit configured to perform a configuration thereof in accordance with the transmitted configuration information

[Fig. 7 illustrates the "Set OM Variable" printer procedure. After the second device (i.e., a selected printer) receives the packet sent from the host computer, the printer controller's Options Manager reads the value of the OM variable (from the data packet) and stores it into memory; **col. 16, lines 32 – 35.**

"Configuring the second device" is achieved by storing the new OM variable values into memory**].**

However, as noted by applicant's remarks filed on 12/20/2007 on **page 11, lines 2 – 4**, GOFFINET *does not disclose or suggest any method or mechanism for determining that the model numbers are the same*.

HANSEN teaches a system and method of retrieving data from a server and cites, "A system includes a server and a controller embedded in a device. Both the server and the embedded controller are capable of communicating over a computer network. The embedded controller sends a command to the server over the computer network that identifies <u>*an instance of the device*</u>. In response, the server identifies the <u>*instance of the device*</u> based on the command, retrieves data that is specific to <u>*the instance of the*</u>

device, and sends the data to the embedded controller over the computer network"; **abstract.**

With reference to **Fig. 2**, HANSEN defines "*instance*" of a device as "the specific identity of device 11 as distinguished from other identical devices. The identification information stored in database 24 identifies the instance of device 11. This identification information includes, but is not limited to, data identifying the type of the device, a common (or "friendly") name for the device, the manufacturer of the device, *the model name of the device*, the model number of the device, the serial number of the device, and a universal unique identifier (UUID) for the device"; **col. 3, lines 33 – 42**.

In addition to storing "identification information", the embedded controller's database **24** stores "operational parameters" and "configuration files"; **col. 3, lines 19 – 22.** "Operational parameters constitute settings and /or control instructions for the device 11, which are implemented by embedded controller 17"; **col. 3, lines 23 – 24.** "A configuration file is a file that contains a set of one or more operational parameters for an instance of device 11"; **col. 3, lines 29 – 31.**

With reference to **Fig. 1**, the server's database **30** stores *operational parameters* "individually or as part of a configuration file for an instance of device 11"; **col. 4, lines 17 – 19.**

With reference to **Fig. 2,** the "embedded controller 17 executes software 20 to retrieve data, such as operational parameters, from remote server 19"; **col. 4, lines 27 – 28.** In step **201**, the embedded controller [i.e., a "second device"] sends a command to the server [i.e., an "information processing apparatus"] which "includes data for identifying device 11. The data identifies the specific instance of device 11 and includes a device type field and one or both of a device serial number field and a device UUID. The command may also include the common name field, the manufacturer name field, *the model name field*, and the model number field"; **col. 4, lines 42 – 46.**

In step **202**, the server "receives the command from embedded controller 17" [**col. 5, lines 39 - 40**] and in step **203**, the server parses "the various identifying fields"; **col. 5, lines 45 – 47.** In step **204**, the server "identifies the instance of device 11 based on the information parsed from the command"; **col. 5, lines 48 – 49.** Referring to **Fig. 5**, in step **205**, data which is specific to the instance of the device is retrieved, and in step **206**, the retrieved data is sent to the embedded controller.

By sending identification information from the embedded controller (i.e., a "second device") to a server (i.e., an "information processing apparatus"), and causing the server to identify the instance of the device (containing the embedded controller) based on the identification information, HANSEN teaches a method and mechanism for determining that the model numbers are the same.

It would have been obvious to one of ordinary skill in the art at the time the invention was made combine the teachings of HANSEN with those of GOFFINET so that the information processing apparatus could *determine whether or not the model information of the first device and the model information of the second device coincide with each other.*

Regarding claim 3, GOFFINET further teaches the device configuring system as claimed in claim 2,

> **wherein the second device further comprises a completion information transmitting unit configured to transmit, after the configuration is completed, completion information that indicates the completion of the configuration to the information processing apparatus**
>
> **[**After the OM configuration variable's value has been set in the second device (i.e., a selected printer; see **Fig. 7,** step **168**), a "success" printer response may be sent back to the host computer;
>
> alternatively, a "failure response will be transmitted if the data size checking failed, or if the [oid1] and [oid2] NPA identification was not acceptable by this particular laser printer"; **col. 16, lines 62 – 65;** in the latter case, an NPA identification may be deemed not acceptable if it is "instructed to change an attribute for a feature not installed on the printer (e.g., if paper tray 3 is being set

to size A4 paper, and a third paper tray is not installed …), it will ignore this Set OM Variable command"; **col. 15, lines 35 - 39].**

Regarding claim 4, GOFFINET further teaches the device configuring system as claimed in claim 2,

wherein the information processing apparatus further comprises an editing unit configured to edit the configuration information

[GOFFINET teaches, "a laser printer should have the capability of having its configuration information contents uploaded into a host computer, so that the host computer can store that same configuration information upon its own storage media, such as in a file residing on a hard disk drive. Once a file is created at the host computer, it will be understood that the contents of such file can either be directly downloaded to the other laser printers on the network, or that the file's contents could be <u>*manipulated*</u> so that individual operating characteristics of a laser printer can be <u>*modified*</u> by a Network Administrator"; **col. 6, lines 33 – 42],**

wherein the storing unit is further configured to store the edited configuration information

[As noted above, the stored configuration file's contents can be "<u>*manipulated*</u> so that individual operating characteristics of a laser printer can be <u>*modified*</u> by a

Network Administrator, and that the file is stored in a storage media (e.g., a hard disk drive)**],**

and wherein the transmitting unit is configured to transmit the edited configuration information as the configuration information to the second device

[As noted above, the configuration file can be either *directly downloaded* (i.e., without modification) to other printers on the network, *or* first modified and then sent to other printers on the network**].**

Regarding claim 6, GOFFINET teaches a device configuring system comprising:

a plurality of devices of various kinds

[GOFFINET teaches "each of the printers 13, 16a, 16b, 16c, and 16d may be of the same type or of different models"; **col. 3, lines 57 - 59];**

and an information processing apparatus [**Fig. 1,** host computer **12**] **which is connected to the devices via a communication network** [**Fig. 1** LAN **15**], **wherein the information processing apparatus comprises:**

a first acquiring unit configured to acquire from a first device [Fig. 1 printer **13] model information of the first device**

[GOFFINET teaches a method by which a host computer (**Fig. 1** host computer **12**) may "save the configuration information of a particular printer (e.g., printer 13)"; **col. 6, lines 47 - 48.** This is illustrated in **Fig. 4** as the "Quick Setup Save" procedure.

Within the printer controller (**Fig. 3**), the "Options Manager 37 is designed to be able to easily retrieve and store all such configuration variables for its particular model laser printer"; **col. 7, lines 5 – 7.**

Shown in **Fig. 4,** steps **102 (**Determine I.D. of Next OM Variable to be Read**)** through **110 (**At End of OM Table?**),** the Options Manager reads each OM variable shown in **Table #1** and transmits the corresponding value to the host computer where it is stored in a file.

From **Table #1,** the host computer acquires "model information" of a first device (e.g., OMMODELNAME, the model name) and "identification information" (e.g., OMSERIALNUM, the serial number);

other related "model information" could be obtained from Table #1 as well; for example, some "higher-end" models may have 3 paper input trays whereas, some "lower-end" models may just have one**];**

a configuration information acquiring unit configured to acquire from the first device configuration information of the first device

[Contained within **Table #1** are various configuration variables (i.e., "OM variables"); among these are, for example, **OMEMULATION** (default emulation), **OMPAPERSRC** (default paper source), **OMOUTPUTCAP** (output drawer capacity), and **OMINPUTCAP** (input tray1 capacity)**];**

a storing unit configured to store the acquired configuration information in a status correlated with the model information of the first device

[Fig. 4 "depicts a flow chart of the steps that the host computer (e.g., a host 12) must undergo to create a file at its own storage media (e.g., upon its own hard disk drive) so as to save the configuration information of a particular printer"; **col. 6, lines 44 – 47;** the storing of the acquired configuration information occurs in **Fig. 4** step **112** (close file in which printer settings are stored)**];**

and a transmitting unit configured to transmit the stored configuration information of the first device together with the correlated model information to a second device

[Fig. 6 illustrates the "Quick Setup Send" host computer procedure. Once a configuration setup file has been selected (**step 140**), a selection is made as to which printers will be configured (**step 142**). **Steps 144** through **152** retrieve configuration values stored in the saved setup file and transmit each value (along

with its corresponding variable identification) to a second device (i.e., a selected printer); specifically, the data packet for a “Set OM Variable” command has a format shown in **col. 15, line 29],**

wherein the second device [e.g., printer **16a** shown in **Fig. 1**, or another printer similar to printer **13** shown in **Fig. 1] comprises:**

a determining unit configured to determine whether or not the transmitted model information of the first device coincides with a previously stored model information thereof

[Each printer stores its configuration variables in NVRAM; **col. 13, lines 58 – 60**;

GOFFINET teaches that each printer may store its configuration variables “in different physical memory locations” (**col. 7, lines 1 - 3**) but that it is the Options Manager’s task to “easily retrieve and store all such configuration variables for its particular model laser printer”; **col. 7, lines 5 – 7.**

GOFFINET teaches that the second device (i.e., a selected printer) has a “determining unit” which “determines whether or not the data is an acceptable value and whether or not the data for a particular attribute (i.e., for an OM

variable) corresponds to the options and configuration" of a printer; **col. 15, lines 32 – 35.**

"Model information" may be defined as the features and installed accessories of a particular device; in this case, a printer may have up to 3 input paper trays. However, "if a particular printer is instructed to change an attribute for a feature not installed on the printer (e.g., if paper tray 3 is being set to size A4 paper, and a third paper tray is not installed …), it will ignore this Set OM Variable command"; **col. 15, lines 35 - 39];**

and a configuring unit configured to perform a configuration thereof in accordance with the transmitted configuration information in a case where determined that the transmitted model information and the previously stored model information coincide each other

[Fig. 7 illustrates the "Set OM Variable" printer procedure. After the second device (i.e., a selected printer) receives the packet sent from the host computer, the printer controller's Options Manager reads the value of the OM variable (from the data packet) and stores it into memory; **col. 16, lines 32 – 35.**

"Configuring the second device" is achieved by storing the new OM variable values into memory**].**

However, as noted by applicant's remarks filed on 12/20/2007 on **page 11, lines 2 – 4**, GOFFINET *does not disclose or suggest any method or mechanism for determining that the model numbers are the same*.

HANSEN teaches a system and method of retrieving data from a server and cites, "A system includes a server and a controller embedded in a device. Both the server and the embedded controller are capable of communicating over a computer network. The embedded controller sends a command to the server over the computer network that identifies *an instance of the device*. In response, the server identifies the *instance of the device* based on the command, retrieves data that is specific to *the instance of the device*, and sends the data to the embedded controller over the computer network"; **abstract.**

With reference to **Fig. 2**, HANSEN defines "*instance*" of a device as "the specific identity of device 11 as distinguished from other identical devices. The identification information stored in database 24 identifies the instance of device 11. This identification information includes, but is not limited to, data identifying the type of the device, a common (or "friendly") name for the device, the manufacturer of the device, *the model name of the device*, the model number of the device, the serial number of the device, and a universal unique identifier (UUID) for the device"; **col. 3, lines 33 – 42**.

In addition to storing “identification information”, the embedded controller’s database **24** stores “operational parameters” and “configuration files”; **col. 3, lines 19 – 22.** “Operational parameters constitute settings and /or control instructions for the device 11, which are implemented by embedded controller 17”; **col. 3, lines 23 – 24.** “A configuration file is a file that contains a set of one or more operational parameters for an instance of device 11”; **col. 3, lines 29 – 31.**

With reference to **Fig. 1**, the server’s database **30** stores *operational parameters* “individually or as part of a configuration file for an instance of device 11”; **col. 4, lines 17 – 19.**

With reference to **Fig. 2,** the “embedded controller 17 executes software 20 to retrieve data, such as operational parameters, from remote server 19”; **col. 4, lines 27 – 28.** In step **201**, the embedded controller [i.e., a “second device”] sends a command to the server [i.e., an “information processing apparatus”] which “includes data for identifying device 11. The data identifies the specific instance of device 11 and includes a device type field and one or both of a device serial number field and a device UUID. The command may also include the common name field, the manufacturer name field, *the model name field*, and the model number field”; **col. 4, lines 42 – 46.**

In step **202**, the server “receives the command from embedded controller 17” [**col. 5, lines 39 - 40**] and in step **203**, the server parses “the various identifying fields”; **col. 5,**

lines 45 – 47. In step **204**, the server “identifies the instance of device 11 based on the information parsed from the command”; **col. 5, lines 48 – 49.** Referring to **Fig. 5**, in step **205**, data which is specific to the instance of the device is retrieved, and in step **206**, the retrieved data is sent to the embedded controller.

By sending identification information from the embedded controller (i.e., a “second device”) to a server (i.e., an “information processing apparatus”), and causing the server to identify the instance of the device (containing the embedded controller) based on the identification information, HANSEN teaches a method and mechanism for determining that the model numbers are the same.

It would have been obvious to one of ordinary skill in the art at the time the invention was made combine the teachings of HANSEN with those of GOFFINET so that the information processing apparatus could *determine whether or not the model information of the first device and the model information of the second device coincide with each other.*

As for claim 7, GOFFINET teaches an information processing apparatus for configuring a plurality of devices of various kinds that are connected thereto via a communication network, the apparatus comprising:

a first acquiring unit configured to acquire from a first device [Fig. 1 printer **13] both model information of the first device and identification information specific to the first device**

[GOFFINET teaches a method by which a host computer (**Fig. 1** host computer **12**) may "save the configuration information of a particular printer (e.g., printer 13)"; **col. 6, lines 47 - 48.** This is illustrated in **Fig. 4** as the "Quick Setup Save" procedure.

Within the printer controller (**Fig. 3**), the "Options Manager 37 is designed to be able to easily retrieve and store all such configuration variables for its particular model laser printer"; **col. 7, lines 5 – 7.**

Shown in **Fig. 4,** steps **102 (**Determine I.D. of Next OM Variable to be Read**)** through **110 (**At End of OM Table?**),** the Options Manager reads each OM variable shown in **Table #1** and transmits the corresponding value to the host computer where it is stored in a file.

From **Table #1,** the host computer acquires "model information" of a first device (e.g., OMMODELNAME, the model name) and "identification information" (e.g., OMSERIALNUM, the serial number);

other related “model information” could be obtained from Table #1 as well; for example, some “higher-end” models may have 3 paper input trays whereas, some “lower-end” models may just have one**];**

a configuration information acquiring unit configured to acquire from the first device configuration information of the first device

[Contained within **Table #1** are various configuration variables (i.e., “OM variables”); among these are, for example, **OMEMULATION** (default emulation), **OMPAPERSRC** (default paper source), **OMOUTPUTCAP** (output drawer capacity), and **OMINPUTCAP** (input tray1 capacity)**];**

a storing unit configured to store the acquired configuration information in a status correlated with both the model information and the identification information of the first device

[Fig. 4 “depicts a flow chart of the steps that the host computer (e.g., a host 12) must undergo to create a file at its own storage media (e.g., upon its own hard disk drive) so as to save the configuration information of a particular printer”; **col. 6, lines 44 – 47;** the storing of the acquired configuration information occurs in **Fig. 4** step **112** (close file in which printer settings are stored)**];**

a second acquiring unit configured to acquire from a second device [e.g., printer **16a** shown in **Fig. 1] both model information of the second device and identification information specific to the second device**

[As with the first device (i.e., printer **13**), the host computer could obtain both model and identification information specific to a second device, and display model information as shown in **Fig. 8;** "those printers having bi-directional communications capabilities are noted with an asterisk, such as that indicated by index numeral 212"; **col. 22, lines 12 - 14];**

a determining unit configured to determine whether or not the model information of the first device and the model information of the second device coincide with each other

[GOFFINET teaches that once the model, identification and configuration information are acquired from the first device and are stored in a file (**Fig. 4** step **112**), "the file on the hard drive can be accessed and its contents sent to other printers on the LAN 15, thereby configuring such other printers very quickly and easily"; **col. 15, lines 3 – 6.**

GOFFINET further teaches that "under normal circumstances, it is preferred that such setup or configuration information for a particular printer be utilized on other printers having the identical model number"; **col. 15, lines 6 - 9];**

and a transmitting unit configured to transmit, when determined that the model information of the first device and the model information of the second device coincide with each other, the stored configuration information of the first device to the second device

[As noted above, GOFFINET teaches that it is preferable that the first and second devices have the “identical model number”.

Fig. 6 illustrates the “Quick Setup Send” host computer procedure. Once a configuration setup file has been selected (**step 140**), a selection is made as to which printers will be configured (**step 142**). **Steps 144** through **152** retrieve configuration values stored in the saved setup file and transmit each value (along with its corresponding variable identification) to a second device (i.e., a selected printer); specifically, the data packet for a “Set OM Variable” command has a format shown in **col. 15, line 29].**

However, as noted by applicant’s remarks filed on 12/20/2007 on **page 11, lines 2 – 4**, GOFFINET *does not disclose or suggest any method or mechanism for determining that the model numbers are the same*.

HANSEN teaches a system and method of retrieving data from a server and cites, “A system includes a server and a controller embedded in a device. Both the server and the embedded controller are capable of communicating over a computer network. The

embedded controller sends a command to the server over the computer network that identifies *an instance of the device*. In response, the server identifies the *instance of the device* based on the command, retrieves data that is specific to *the instance of the device*, and sends the data to the embedded controller over the computer network"; **abstract.**

With reference to **Fig. 2**, HANSEN defines "*instance*" of a device as "the specific identity of device 11 as distinguished from other identical devices. The identification information stored in database 24 identifies the instance of device 11. This identification information includes, but is not limited to, data identifying the type of the device, a common (or "friendly") name for the device, the manufacturer of the device, *the model name of the device*, the model number of the device, the serial number of the device, and a universal unique identifier (UUID) for the device"; **col. 3, lines 33 – 42**.

In addition to storing "identification information", the embedded controller's database **24** stores "operational parameters" and "configuration files"; **col. 3, lines 19 – 22.** "Operational parameters constitute settings and /or control instructions for the device 11, which are implemented by embedded controller 17"; **col. 3, lines 23 – 24.** "A configuration file is a file that contains a set of one or more operational parameters for an instance of device 11"; **col. 3, lines 29 – 31.**

With reference to **Fig. 1**, the server's database **30** stores *operational parameters* "individually or as part of a configuration file for an instance of device 11"; **col. 4, lines 17 – 19.**

With reference to **Fig. 2,** the "embedded controller 17 executes software 20 to retrieve data, such as operational parameters, from remote server 19"; **col. 4, lines 27 – 28.** In step **201**, the embedded controller [i.e., a "second device"] sends a command to the server [i.e., an "information processing apparatus"] which "includes data for identifying device 11. The data identifies the specific instance of device 11 and includes a device type field and one or both of a device serial number field and a device UUID. The command may also include the common name field, the manufacturer name field, *the model name field*, and the model number field"; **col. 4, lines 42 – 46.**

In step **202**, the server "receives the command from embedded controller 17" [**col. 5, lines 39 - 40**] and in step **203**, the server parses "the various identifying fields"; **col. 5, lines 45 – 47.** In step **204**, the server "identifies the instance of device 11 based on the information parsed from the command"; **col. 5, lines 48 – 49.** Referring to **Fig. 5**, in step **205**, data which is specific to the instance of the device is retrieved, and in step **206**, the retrieved data is sent to the embedded controller.

By sending identification information from the embedded controller (i.e., a "second device") to a server (i.e., an "information processing apparatus"), and causing the server

to identify the instance of the device (containing the embedded controller) based on the identification information, HANSEN teaches a method and mechanism for determining that the model numbers are the same.

It would have been obvious to one of ordinary skill in the art at the time the invention was made combine the teachings of HANSEN with those of GOFFINET so that the information processing apparatus could *determine whether or not the model information of the first device and the model information of the second device coincide with each other.*

As for claim 8, GOFFINET teaches ~~a software program product for~~ a computer readable medium storing a program causing a computer system to execute ~~procedures~~ a process for configuring a plurality of devices of various kinds that are connected thereto via a communication network, ~~the software program product~~ the process comprising:

> ~~**means for**~~ **acquiring, from a first device [Fig. 1** printer **13], model information of the first device, identification information specific to the first device**
>
> **[**GOFFINET teaches a method by which a host computer (**Fig. 1** host computer **12**) may "save the configuration information of a particular printer (e.g., printer 13)"; **col. 6, lines 47 - 48.** This is illustrated in **Fig. 4** as the "Quick Setup Save" procedure.

Within the printer controller (**Fig. 3**), the “Options Manager 37 is designed to be able to easily retrieve and store all such configuration variables for its particular model laser printer”; **col. 7, lines 5 – 7.**

Shown in **Fig. 4,** steps **102 (**Determine I.D. of Next OM Variable to be Read**)** through **110 (**At End of OM Table?**),** the Options Manager reads each OM variable shown in **Table #1** and transmits the corresponding value to the host computer where it is stored in a file.

From **Table #1,** the host computer acquires “model information” of a first device (e.g., OMMODELNAME, the model name) and “identification information” (e.g., OMSERIALNUM, the serial number);

other related “model information” could be obtained from Table #1 as well; for example, some “higher-end” models may have 3 paper input trays whereas, some “lower-end” models may just have one**],**

and configuration information of the first device

[Contained within **Table #1** are various configuration variables (i.e., “OM variables”); among these are, for example, **OMEMULATION** (default emulation), **OMPAPERSRC** (default paper source), **OMOUTPUTCAP** (output drawer capacity), and **OMINPUTCAP** (input tray1 capacity)**;**

~~**means for**~~ **storing the acquired configuration information in a status correlated with both the model information and the identification information of the first device**

[Fig. 4 "depicts a flow chart of the steps that the host computer (e.g., a host 12) must undergo to create a file at its own storage media (e.g., upon its own hard disk drive) so as to save the configuration information of a particular printer"; **col. 6, lines 44 – 47;** the storing of the acquired configuration information occurs in **Fig. 4** step **112** (close file in which printer settings are stored)**];**

~~**means for**~~ **acquiring from a second device [**e.g., printer **16a** shown in **Fig. 1]** **both model information of the second device and identification information specific to the second device**

[As with the first device (i.e., printer **13**), the host computer could obtain both model and identification information specific to a second device, and display model information as shown in **Fig. 8;** "those printers having bi-directional communications capabilities are noted with an asterisk, such as that indicated by index numeral 212"; **col. 22, lines 12 - 14];**

~~**means for determining**~~ <u>**causing the information processing apparatus or the second device to determine**</u> **whether or not the model information of the**

first device and the model information of the second device coincide with each other

[GOFFINET teaches that once the model, identification and configuration information are acquired from the first device and are stored in a file (**Fig. 4** step **112**), "the file on the hard drive can be accessed and its contents sent to other printers on the LAN 15, thereby configuring such other printers very quickly and easily"; **col. 15, lines 3 – 6.**

GOFFINET further teaches that "under normal circumstances, it is preferred that such setup or configuration information for a particular printer be utilized on other printers having the identical model number"; **col. 15, lines 6 - 9];**

and ~~means for~~ transmitting, when determined that the model information of the first device and the model information of the second device coincide with each other, the stored configuration information of the first device to the second device

[As noted above, GOFFINET teaches that it is preferable that the first and second devices have the "identical model number".

Fig. 6 illustrates the "Quick Setup Send" host computer procedure. Once a configuration setup file has been selected (**step 140**), a selection is made as to which printers will be configured (**step 142**). **Steps 144** through **152** retrieve

configuration values stored in the saved setup file and transmit each value (along with its corresponding variable identification) to a second device (i.e., a selected printer); specifically, the data packet for a "Set OM Variable" command has a format shown in **col. 15, line 29].**

However, as noted by applicant's remarks filed on 12/20/2007 on **page 11, lines 2 – 4**, GOFFINET *does not disclose or suggest any method or mechanism for determining that the model numbers are the same*.

HANSEN teaches a system and method of retrieving data from a server and cites, "A system includes a server and a controller embedded in a device. Both the server and the embedded controller are capable of communicating over a computer network. The embedded controller sends a command to the server over the computer network that identifies *<u>an instance of the device</u>*. In response, the server identifies the *<u>instance of the device</u>* based on the command, retrieves data that is specific to *<u>the instance of the device</u>*, and sends the data to the embedded controller over the computer network"; **abstract.**

With reference to **Fig. 2**, HANSEN defines "*<u>instance</u>*" of a device as "the specific identity of device 11 as distinguished from other identical devices. The identification information stored in database 24 identifies the instance of device 11. This identification information includes, but is not limited to, data identifying the type of the device, a common (or

“friendly”) name for the device, the manufacturer of the device, *the model name of the device*, the model number of the device, the serial number of the device, and a universal unique identifier (UUID) for the device”; **col. 3, lines 33 – 42**.

In addition to storing “identification information”, the embedded controller’s database **24** stores “operational parameters” and “configuration files”; **col. 3, lines 19 – 22.** “Operational parameters constitute settings and /or control instructions for the device 11, which are implemented by embedded controller 17”; **col. 3, lines 23 – 24.** “A configuration file is a file that contains a set of one or more operational parameters for an instance of device 11”; **col. 3, lines 29 – 31.**

With reference to **Fig. 1**, the server’s database **30** stores *operational parameters* “individually or as part of a configuration file for an instance of device 11”; **col. 4, lines 17 – 19.**

With reference to **Fig. 2,** the “embedded controller 17 executes software 20 to retrieve data, such as operational parameters, from remote server 19”; **col. 4, lines 27 – 28.** In step **201**, the embedded controller [i.e., a “second device”] sends a command to the server [i.e., an “information processing apparatus”] which “includes data for identifying device 11. The data identifies the specific instance of device 11 and includes a device type field and one or both of a device serial number field and a device UUID. The

command may also include the common name field, the manufacturer name field, *the model name field*, and the model number field"; **col. 4, lines 42 – 46.**

In step **202**, the server "receives the command from embedded controller 17" [**col. 5, lines 39 - 40**] and in step **203**, the server parses "the various identifying fields"; **col. 5, lines 45 – 47.** In step **204**, the server "identifies the instance of device 11 based on the information parsed from the command"; **col. 5, lines 48 – 49.** Referring to **Fig. 5**, in step **205**, data which is specific to the instance of the device is retrieved, and in step **206**, the retrieved data is sent to the embedded controller.

By sending identification information from the embedded controller (i.e., a "second device") to a server (i.e., an "information processing apparatus"), and causing the server to identify the instance of the device (containing the embedded controller) based on the identification information, HANSEN teaches a method and mechanism for determining that the model numbers are the same.

It would have been obvious to one of ordinary skill in the art at the time the invention was made combine the teachings of HANSEN with those of GOFFINET so that the information processing apparatus could *determine whether or not the model information of the first device and the model information of the second device coincide with each other.*

7. Claim 5 is rejected under 35 U.S.C. 103(a) as being unpatentable over **GOFFINET [US Patent 5,905,906]** and **HANSEN [US Patent 7,185,014 B1]** in view of **TATEYAMA [US Patent 6,425,019 B1]**.

Regarding claim 5, neither GOFFINET nor HANSEN specifically teach the device configuring system as claimed in claim 2,

wherein the identification comprises an MAC address of the device.

TATEYAMA teaches a data communications method among various types of devices which may include computers, printers, and storage devices; **col. 5, lines 62 – col. 6, line 3.** TATEYAMA further teaches the identifier (ID) unique to each device "may be a network address such as an Internet Protocol (IP) address or a Media Access Control (MAC) address"; **col. 22, lines 40 – 42.**

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of TATEYAMA with those of GOFFINET and HANSEN to use a network device's MAC address as an identification since network devices are assigned *unique* and specific MAC addresses at the time of manufacture.

Conclusion

8. The prior art made of record and not relied upon is considered pertinent to

applicant's disclosure:

- **KAI [US Patent 6,587,971 B1]** teaches a method in which a host computer obtains the model name of a printer and then compares it "*with the model name of a printer which is assumed by the image drawing commands*"; **col. 13, lines 13 – 16.** When the printer model names coincide, the image drawing commands are sent to the printer; otherwise, the image drawing commands are not sent. In this way, "*waste of print paper, ink, toner*" can be prevented; **col. 13, lines 51 – 56.** With reference to **Fig. 16**, KAI further teaches a "*printer information acquisition module*" which is a "*program module for acquiring the printer type information*", a "*printer model name comparison module*" which is "*a program for comparing the model name of the printer included in the printer type information*" and a "*command transmission control module*" which is "*a program module for controlling whether the image drawing commands are transmitted or not in accordance with the result of comparison of the model names*"; **col. 17, line 60 – col. 18, line 4.**

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Peter L. Cheng whose telephone number is 571-270-3007. The examiner can normally be reached on MONDAY - FRIDAY, 8:30 AM - 6:00 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, King Y. Poon can be reached on 571-272-7440. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/King Y. Poon/
Supervisory Patent Examiner, Art Unit 2625

plc
May 5, 2008